# RIS-Assisted Jamming Rejection and Path Planning for UAV-Borne IoT Platform: A New Deep Reinforcement Learning Framework

Shuyan Hu, *Member, IEEE*, Xin Yuan, *Member, IEEE*, Wei Ni, *Senior Member, IEEE*,
Xin Wang, *Fellow, IEEE*, and Abbas Jamalipour, *Fellow, IEEE*

*Abstract*—This paper presents a new deep reinforcement learning (DRL)-based approach to the trajectory planning and jamming rejection of an unmanned aerial vehicle (UAV) for the Internet-of-Things (IoT) applications. Jamming can prevent timely delivery of sensing data and reception of operation instructions. With the assistance of a reconfigurable intelligent surface (RIS), we propose to augment the radio environment, suppress jamming signals, and enhance the desired signals. The UAV is designed to learn its trajectory and the RIS configuration based solely on changes in its received data rate, using the latest deep deterministic policy gradient (DDPG) and twin delayed DDPG (TD3) models. Simulations show that the proposed DRL algorithms give the UAV with strong resistance against jamming and that the TD3 algorithm exhibits faster and smoother convergence than the DDPG algorithm, and suits better for larger RISs. This DRL-based approach eliminates the need for knowledge of the channels involving the RIS and jammer, thereby offering significant practical value.

*Index Terms*—Internet of Things, unmanned aerial vehicle, jamming rejection, reconfigurable intelligent surface, deep deterministic policy gradient (DDPG), twin delayed DDPG (TD3).

## I. INTRODUCTION

The use of unmanned aerial vehicles (UAVs) has been growing in popularity with the expansion of the Internet of Things (IoT) [1]–[3]. For instance, UAVs equipped with sensors and cameras are increasingly deployed to monitor and collect data on air quality, traffic, and other environmental factors for IoT applications [4], [5]. The data collected by the UAVs need to be transmitted back to a control center or ground base station (BS) for data analysis or for triggering automated responses, such as adjusting traffic lights to alleviate congestion [6].

Jamming can have severe effects on the physical-layer security of UAV-borne IoT applications [7]–[10]. An attacker can use jamming equipment to disrupt the wireless transmissions between the UAVs and the BS, preventing the UAVs from transmitting their sensing data and receiving operation

instructions in a timely manner and causing the UAVs to fail their missions [11]. As a matter of fact, jamming has been identified to be one of the most critical threats to the massive IoT connections in the context of the upcoming sixth-generation (6G) communication systems [12].

Identifying and eliminating jammers is challenging without specialized equipment, such as radar [13]. Previous research on jamming cancellation for UAV-borne IoT applications has not specifically focused on UAV-borne IoT platforms. In [14], the authors targeted to improve the worst-case long-term data rate for a UAV-assisted wireless sensor network under jamming attacks by jointly configuring the UAV's transmission strategy and 3D flight path. In [15], a reinforcement learning (RL)-based communication strategy was developed between a UAV swarm and a BS to counteract jamming attempts. This improved the communication quality of UAVs by exploring the motion and antenna spatial domain. In [16], the authors proposed a secure UAV communication scheme against smart jammers using a knowledge-based RL approach, which leveraged domain information to reduce the state space and speed up the convergence of the RL algorithm.

On the other hand, reconfigurable intelligent surfaces (RISs) were developed as a means of creating programmable wireless transmission environments [17]. These surfaces, made up of passive reflecting units with reprogrammable phase shifts [18], can be placed on building surfaces, and are expected to be an integral part and effective enhancer for the IoT. By configuring the phase shifts, the RIS-reflected signals are added constructively to the direct signal to improve signal quality or destructively to reduce interference [19]. However, the potential of RISs for anti-jamming applications has not been widely studied [20], [21]. In [22], a collective active and passive beamforming design was formulated to lower transmit power by optimizing the continuous phase shifts of an RIS. The result was later extended to discrete phase shifts in [23]. The RIS-assisted secure transmission was studied in [24], [25], where fast RL and deep RL (DRL) were used to design the active and passive beamformers. For instance, in [24], jamming signals were modulated and reflected by an RIS to reduce the eavesdropping data rate.

As far as we know, the RIS-assisted jamming rejection has yet to be addressed in the context of UAV-borne IoT platforms. While there have been studies that have examined general RIS-assisted UAV communications, e.g., in [26]–[30], none have considered the impact of jamming attacks. For example,

Shuyan Hu and Xin Yuan contributed equally to this work.

S. Hu and X. Wang are with the Department of Communication Science and Engineering, Fudan University, Shanghai 200433, China (e-mails: {syhu14, xwang11}@fudan.edu.cn).

X. Yuan and W. Ni are with CSIRO Data61, Sydney, NSW 2122, Australia (e-mails: {xin.yuan, wei.ni}@data61.csiro.au).

A. Jamalipour is with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia (email: a.jamalipour@ieee.org).

Corresponding author: X. Wang.

in [27], the authors jointly optimized UAV flight path and RIS passive beamformers to achieve the largest average rate of the terrestrial user. In [29], a UAV and RIS were configured to deliver ultra-reliable and low-latency commands among terrestrial IoT devices using nonconvex optimization. These studies are inapplicable to the RIS-assisted jamming rejection for UAV-borne IoT platforms.

In this paper, we put forth a new DRL-based architecture for the flight path planning and jamming cancellation of UAV-borne IoT platforms. A fixed-wing UAV is considered. Our architecture utilizes an RIS, which dynamically modifies the wireless transmission environment to mitigate jamming power and enhance intended signals to the UAV. To accomplish this, we devise a new Deep Deterministic Policy Gradient (DDPG) model and its enhancement, Twin-Delayed DDPG (TD3), to allow the UAV to learn its flight path and the RIS configuration based solely on its received data rate, eliminating the need for the channel state information (CSI) involving the RIS and jammer in the flight path training. This presents a significant practical advantage, as the estimation of CSI involving an RIS is complex and may not be able to be performed in real-time.

The main contributions of this paper are as follows:

- A new problem is introduced to jointly optimize flight path planning and RIS-assisted jamming cancellation to maximize the data rate of a UAV-borne IoT platform. The problem is non-straightforward for its non-convexity and sequential decision-making nature.
- A new DRL architecture is proposed to solve the new problem and allow the UAV to learn its flight path and the RIS configuration based solely on its received data rate, eliminating the need for CSI in the training process.
- The DRL architecture is implemented using the latest DRL models, DDPG and its twin-delayed version, i.e., TD3. While TD3 is generally applicable to the problem under investigation, DDPG can benefit from its simpler network architecture and smooth convergence and is suitable for problems with smaller scales, e.g., fewer RISs, and less stringent mission time requirements.

The proposed DRL approach is validated through extensive simulations, showing its exceptional resistance against jamming. Particularly, the DDPG demonstrates faster and smoother convergence when the mission time is long and the RIS is small. The TD3 outperforms the DDPG in robustness against the jammer's position, especially when the mission time is short. This is crucial, as accurately locating the jammer and estimating its CSI can be practically challenging.

The remainder of this paper is arranged as follows. In Section II, the system model is described. In Section III, we formulate the problem of jointly designing the UAV's flight path and the RIS configuration to maximize the data rate in the presence of an unknown jammer. In Section IV, we propose new DRL solutions to the problem. Performances are gauged in Section V. The paper is concluded in Section VI.

*Notation*: Boldface lower- and upper-cases indicate vectors and matrices, respectively; $\mathbb{C}^N$ denotes the space of $N \times 1$ complex-valued column-vectors; $\|\cdot\|$ denotes the Euclidean norm; $(\cdot)^T$ and $(\cdot)^H$ stand for transpose and conjugate transpose, respectively; $\text{diag}(\mathbf{a})$ is a diagonal matrix with the



Fig. 1. An RIS-assisted anti-jamming UAV-borne IoT platform, where an RIS is adaptively configured along with the UAV trajectory to enhance the desired signals and reject the jamming signals.

elements of $\mathbf{a}$ along the diagonal; $\otimes$ stands for the Kronecker product.

## II. SYSTEM MODEL

As depicted in Fig. 1, a ground BS offers wireless communication service for a fixed-wing UAV in the presence of a terrestrial jammer. Consider a three-dimensional (3D) Cartesian coordinate system. The BS and the jammer are placed at $\mathbf{q}_{\mathrm{B}} = [0,0,0]^T$ and $\mathbf{q}_{\mathrm{J}} = [x_{\mathrm{J}}, y_{\mathrm{J}}, 0]^T$. An RIS is deployed to facilitate the UAV communication and suppress the jamming signals from the jammer. We suppose that the BS, UAV and jammer all have a single antenna for description convenience (but the proposed DRL architecture can be extended to support multi-antenna BSs and UAVs, as part of our future work). Only based on its received data rate, the UAV determines its flight path and the RIS configuration dynamically. The RIS is installed with a smart controller to procure the command from the UAV (via the cellular system) for RIS configuration [31].

### A. UAV Mobility Model

The UAV functions for a finite scheduling horizon of $T$ seconds, which is split into $T_w$ time slots indexed by $t$, and $t = 1, \cdots, T_w$. A slot lasts $\delta = T/T_w$, which is short enough that the UAV can be viewed as stationary per slot. The UAV is traveling from a predetermined starting position $\mathbf{q}_0 = [x_0, y_0, z_0]^T$ to a predetermined ending position $\mathbf{q}_F = [x_F, y_F, z_F]^T$. The UAV's 3D coordinates are $\mathbf{q}_t = [x_t, y_t, z_t]^T, \forall t$. Let $\mathbf{V}_t := [V_{xt}, V_{yt}, V_{zt}]^T$ and $\mathbf{A}_t := [A_{xt}, A_{yt}, A_{zt}]^T$ collect the velocity and acceleration of the UAV per slot $t$, respectively. The UAV obeys some mobility constraints [32]:

$$\mathbf{q}_{t+1} = \mathbf{q}_t + \mathbf{V}_t\delta + \frac{1}{2}\mathbf{A}_t\delta^2, \ \forall t, \tag{1a}$$

$$\mathbf{q}_F = \mathbf{q}_{T_w}, \tag{1b}$$

$$\boldsymbol{V}_{t+1} = \boldsymbol{V}_t + \boldsymbol{A}_t \delta, \ \forall t, \tag{1c}$$

$$|A_{i,t}| \le A_{\max}, i \in \{x, y, z\}, \ \forall t, \tag{1d}$$

$$V_{\min} \le \|\boldsymbol{V}_t\| \le V_{\max}, \ \forall t, \tag{1e}$$

$$V_{zt}/\|\boldsymbol{V}_t\| \le \sin \vartheta, \ \forall t, \tag{1f}$$

where $A_{\max}$ is the UAV's largest acceleration; $V_{\max}$ and $V_{\min}$ are the UAV's maximum and minimum speeds; and $\vartheta$ is the largest UAV pitch angle when ascending or descending.

### B. RIS Configuration

The RIS is adhered on the outer surface of a building, which is on the $(x, z)$-plane and aligns with the $x$-axis. The RIS has a uniform rectangular array (URA) of $N = N_x N_y$ reflecting elements (units), and a controller to dynamically control the phase shift of each unit. Let $\boldsymbol{\Theta}_t := \mathrm{diag}(e^{j\theta_1^t}, \ldots, e^{j\theta_N^t})$ be the phase shift matrix for the RIS at time slot $t$, where $\theta_n^t = \theta_{(n_x-1)N_y+n_y}^t \in [-\pi, \pi), n = 1, \ldots, N$, is the phase shift of the $n$-th reflecting unit which is located at the $n_y$-th row and the $n_x$-th column of the RIS, and $j = \sqrt{-1}$. The first element of the RIS is at the right bottom corner of the RIS, and its coordinates are $\boldsymbol{q}_{\mathrm{R}} = [x_{\mathrm{R}}, y_{\mathrm{R}}, z_{\mathrm{R}}]^T$.

### C. Channel Model

The distances of the BS-UAV link $d_{\mathrm{BU}}^t$, the BS-RIS link $d_{\mathrm{BR}}$, the Jammer-RIS link $d_{\mathrm{JR}}$, the Jammer-UAV link $d_{\mathrm{JU}}^t$, and RIS-UAV link $d_{\mathrm{RU}}^t$ are given by

$$d_{\mathrm{BU}}^t = \|\boldsymbol{q}_t\|, \ \forall t, \ d_{\mathrm{BR}} = \|\boldsymbol{q}_{\mathrm{R}}\|, \ d_{\mathrm{JR}} = \|\boldsymbol{q}_{\mathrm{J}} - \boldsymbol{q}_{\mathrm{R}}\|, \tag{2a}$$

$$d_{\mathrm{JU}}^t = \|\boldsymbol{q}_t - \boldsymbol{q}_{\mathrm{J}}\|, \ d_{\mathrm{RU}}^t = \|\boldsymbol{q}_t - \boldsymbol{q}_{\mathrm{R}}\|, \ \forall t. \tag{2b}$$

Consider an LoS channel for the RIS-UAV link (i.e., the R-U link), and Rician fading channels between the BS/jammer and the UAV (i.e., the B-U and J-U links), and between the BS/jammer and the RIS (i.e., the B-R and J-R links). The channel gains of the B-U and J-U links are: $\forall t$,

$$h_{\mathrm{BU}}^t = \sqrt{\rho(d_{\mathrm{BU}}^t)^{-\kappa_1}} \left( \sqrt{\frac{\beta_t}{1+\beta_t}} g_{\mathrm{BU}}^t + \sqrt{\frac{1}{1+\beta_t}} \tilde{g}_{\mathrm{BU}}^t \right), \tag{3a}$$

$$h_{\mathrm{JU}}^t = \sqrt{\rho(d_{\mathrm{JU}}^t)^{-\kappa_1}} \left( \sqrt{\frac{\beta_t}{1+\beta_t}} g_{\mathrm{JU}}^t + \sqrt{\frac{1}{1+\beta_t}} \tilde{g}_{\mathrm{JU}}^t \right), \tag{3b}$$

where $\rho$ stands for the path loss at the reference distance $d_0 = 1$ m with the path loss exponent $\kappa_1 > 2$; $\beta_t$ is the Rician factor of the B-U and J-U links; $g_{\mathrm{BU}}^t$ and $g_{\mathrm{JU}}^t$ are the deterministic LoS components with $|g_{\mathrm{BU}}^t| = 1$ and $|g_{\mathrm{JU}}^t| = 1$; $\tilde{g}_{\mathrm{BU}}^t$ and $\tilde{g}_{\mathrm{JU}}^t$ stand for stochastic dispersion captured by a zero-mean, unit-variance circularly symmetric complex Gaussian (CSCG) random variable. The elevation-angle-reliant Rician factor $\beta_t$ is captured by the following exponential function [33]

$$\beta_t = \xi_1 \exp \left( \xi_2 \arcsin(z_t/d_{\mathrm{BU}}^t) \right), \ \forall t, \tag{4}$$

where $\xi_1$ and $\xi_2$ are two constant coefficients dependent on the environment.

The channel gains of the B-R, J-R, and R-U links, denoted by $\boldsymbol{h}_{\mathrm{BR}} \in \mathbb{C}^{N \times 1}$, $\boldsymbol{h}_{\mathrm{JR}} \in \mathbb{C}^{N \times 1}$, and $\boldsymbol{h}_{\mathrm{RU}}^t \in \mathbb{C}^{N \times 1}$, are

$$\boldsymbol{h}_{\mathrm{BR}} = \underbrace{\sqrt{\rho d_{\mathrm{BR}}^{-\kappa_2}}}_{\text{path loss}} \underbrace{\left( \sqrt{\frac{\beta}{1+\beta}} \boldsymbol{h}_{\mathrm{BR}}^{los} + \sqrt{\frac{1}{1+\beta}} \boldsymbol{h}_{\mathrm{BR}}^{nlos} \right)}_{\text{array response \& small-scale fading}}; \tag{5a}$$

$$\boldsymbol{h}_{\mathrm{JR}} = \underbrace{\sqrt{\rho d_{\mathrm{JR}}^{-\kappa_2}}}_{\text{path loss}} \underbrace{\left( \sqrt{\frac{\beta}{1+\beta}} \boldsymbol{h}_{\mathrm{JR}}^{los} + \sqrt{\frac{1}{1+\beta}} \boldsymbol{h}_{\mathrm{JR}}^{nlos} \right)}_{\text{array response \& small-scale fading}}; \tag{5b}$$

$$\boldsymbol{h}_{\mathrm{RU}}^t = \sqrt{\rho(d_{\mathrm{RU}}^t)^{-2}} \boldsymbol{g}_{\mathrm{RU}}^t, \ \forall t. \tag{5c}$$

Here, $\beta$ is the Rician factor of the B-R and J-R links (c.f. $\beta_t$); and $\boldsymbol{h}_{\mathrm{BR}}^{los}$ and $\boldsymbol{h}_{\mathrm{JR}}^{los}$ are the LoS components, as given by

$$\boldsymbol{h}_{\mathrm{BR}}^{los} = \left[ 1, \ldots, e^{-j\frac{2\pi d_x}{\lambda}(N_x-1)\phi_{\mathrm{BR}}^x} \right]^T \otimes$$
$$\left[ 1, \ldots, e^{-j\frac{2\pi d_y}{\lambda}(N_y-1)\phi_{\mathrm{BR}}^y} \right]^T, \tag{6a}$$

$$\boldsymbol{h}_{\mathrm{JR}}^{los} = \left[ 1, \ldots, e^{-j\frac{2\pi d_x}{\lambda}(N_x-1)\phi_{\mathrm{JR}}^x} \right]^T \otimes$$
$$\left[ 1, \ldots, e^{-j\frac{2\pi d_y}{\lambda}(N_y-1)\phi_{\mathrm{JR}}^y} \right]^T, \tag{6b}$$

where $d_x$ and $d_y$ are the antenna spacings in the directions of the $x$- and $y$-axes, respectively; $\phi_{\mathrm{BR}}^x = x_{\mathrm{R}}/d_{\mathrm{BR}}$ and $\phi_{\mathrm{BR}}^y = y_{\mathrm{R}}/d_{\mathrm{BR}}$ are the spatial frequencies corresponding to the angles-of-arrivals (AoAs) from BS to RIS, and $\phi_{\mathrm{JR}}^x = (x_{\mathrm{J}} - x_{\mathrm{R}})/d_{\mathrm{JR}}$ and $\phi_{\mathrm{JR}}^y = (y_{\mathrm{J}} - y_{\mathrm{R}})/d_{\mathrm{JR}}$ are the spatial frequencies corresponding to the AoAs from the jammer to the RIS along the $x$- and $y$-axes, respectively [34].

In (5a) and (5b), $\boldsymbol{h}_{\mathrm{BR}}^{nlos} \in \mathbb{C}^{N \times 1}$ and $\boldsymbol{h}_{\mathrm{JR}}^{nlos} \in \mathbb{C}^{N \times 1}$ are the non-LoS (NLoS) components with the variables independently drawn from the zero-mean, unit-variance CSCG distribution. In (5c), $\boldsymbol{g}_{\mathrm{RU}}^t$ is the array response, as given by

$$\boldsymbol{g}_{\mathrm{RU}}^t = \left[ 1, \ldots, e^{-j\frac{2\pi d_x}{\lambda}(N_x-1)\phi_{\mathrm{RU},x}^t} \right]^T \otimes$$
$$\left[ 1, \ldots, e^{-j\frac{2\pi d_y}{\lambda}(N_y-1)\phi_{\mathrm{RU},y}^t} \right]^T, \ \forall t, \tag{7}$$

where $\phi_{\mathrm{RU},x}^t = (x_t - x_{\mathrm{R}})/d_{\mathrm{RU}}^t$ and $\phi_{\mathrm{RU},y}^t = (y_t - y_{\mathrm{R}})/d_{\mathrm{RU}}^t$ are the spatial frequencies corresponding to the angles-of-departures (AoDs) from RIS to UAV along the $x$- and $y$-axes, respectively.

It is noteworthy that the UAV does not ask for the CSI involving the RIS and the jammer to produce its flight path and configure the RIS in this paper. Instead, the UAV only measures its own received data rate to learn the control policy of its flight path and RIS configuration. This consideration is of practical value, since the RIS-reflected channels are difficult and slow to estimate.

## III. PROBLEM FORMULATION

Let $P_t, \forall t$ denote the transmit power of the BS and $P_{\mathrm{J}}$ denote the transmit power of the jammer. The signal-to-interference-plus-noise ratio (SINR) at the UAV at time slot $t$, denoted by $\gamma_{\mathrm{U}}^t$, is

$$\gamma_{\mathrm{U}}^t = \frac{P_t |h_{\mathrm{BU}}^t + (\boldsymbol{h}_{\mathrm{RU}}^t)^H \boldsymbol{\Theta}_t \boldsymbol{h}_{\mathrm{BR}}|^2}{P_{\mathrm{J}} |h_{\mathrm{JU}}^t + (\boldsymbol{h}_{\mathrm{RU}}^t)^H \boldsymbol{\Theta}_t \boldsymbol{h}_{\mathrm{JR}}|^2 + \sigma_{\mathrm{U}}^2}, \tag{8}$$

where $\sigma_{\mathrm{U}}^2$ is the variance of the additive white Gaussian noise (AWGN) at the UAV. The received data rate of the UAV at time slot $t$ is given by

$$R_{\mathrm{U}}^t = \log_2(1 + \gamma_{\mathrm{U}}^t). \tag{9}$$

We aim to maximize the total received data rate of the UAV from the BS over the mission duration of $T_w$ slots. The problem considered is stated as follows.

$$\max_{\{\boldsymbol{q}_t, \boldsymbol{\Theta}_t, \forall t\}} \sum_{t=1}^{T_w} R_{\mathrm{U}}^t \tag{10a}$$

$$\text{s.t.} \ -\pi \le \theta_n^t < \pi, \ \forall n, t, \tag{10b}$$

$$(1a) - (1f). $$

Problem (10) is challenging for traditional convex solvers due to several reasons: First of all, the received data rate is a non-convex function of the UAV's flight path $\boldsymbol{q}_t, \forall t$ and the RIS phase shifts $\boldsymbol{\Theta}_t, \forall t$. Second, the UAV flight path waypoints are embedded in the exponents of the R-U link in (5c) and (7), making the trajectory optimization intractable for existing convex tools, such as successive convex approximation. Another reason is that the large number of RIS reflecting units can cause prohibitive overhead and complexity for radio channel estimation, acquisition and reconfiguration. To overcome these limitations, the next section proposes using DRL to solve (10).

## IV. PROPOSED DRL FRAMEWORK FOR ANTI-JAMMING COMMUNICATION OF UAV-BORNE IoT PLATFORMS

The proposed method in this section aims to solve problem (10) by utilizing the DDPG and TD3 models. Our approach involves learning to adjust the RIS and control the UAV's trajectory, including heading and acceleration, based on changes in the received data rate of the UAV. Importantly, our method eliminates the need for precise CSI or knowledge of the RIS reflecting channels. DDPG and its variations, such as TD3, have been demonstrated to be effective in addressing problems with continuous action spaces [35]–[37]. In contrast, traditional DRL methods, such as deep Q-learning, can struggle and even diverge when faced with continuous action spaces.

### A. State, Action, and Reward

Since the current UAV location only depends on its previous location and speed, the UAV trajectory (i.e., waypoints) is a Markov decision process (MDP). The RIS configuration depends solely on the instantaneous position of the UAV. Therefore, we interpret problem (10) as an MDP with its state, action, and reward defined below.

- **State Space $\mathcal{S}$:** At time slot $t$, the system state $s_t \in \mathcal{S}$ is made of the relative position of the UAV with regards to its final location, $\boldsymbol{q}_t - \boldsymbol{q}_F$, the velocity of the UAV, $\boldsymbol{V}_t$, and the SINR at the UAV, $\gamma_{\mathrm{U}}^t$, $s_t = \{\boldsymbol{q}_t - \boldsymbol{q}_F, V_t, \gamma_{\mathrm{U}}^t\}$.
- **Action Space $\mathcal{A}$:** It gathers all possible actions, i.e., $a_t \in \mathcal{A}$. During the $t$-th time step, the action $a_t$ consists of the reflecting coefficients $\{\theta_n^t\}_{n \in \mathcal{N}}$ and the acceleration of the UAV, $\boldsymbol{A}_t := [A_{xt}, A_{yt}, A_{zt}]^T$, i.e., $a_t = \{\theta_n^t \in [-\pi, \pi), \forall n, A_{it} \in [-A_{\max}, A_{\max}], i \in \{x, y, z\}\}$.

The UAV acceleration is constrained by (1d)–(1f). Given the initial location and velocity of the UAV, its future waypoints $\boldsymbol{q}_t$ and velocities $\boldsymbol{V}_t$ are decided by the accelerations, i.e., by (1a)–(1c).

- **Reward $r_t$:** The reward function gives positive returns per time step for implementing action $a_t$:

$$r_t = \underbrace{R_{\mathrm{U}}^t}_{\text{communication}} + \underbrace{\zeta \left(d_F^{t-1} - d_F^t\right)}_{\text{distance to the final location}}, \tag{11}$$

where $d_F^{t-1} = \|\boldsymbol{q}_{t-1} - \boldsymbol{q}_F\|$ and $d_F^t = \|\boldsymbol{q}_t - \boldsymbol{q}_F\|$ are the distances from the UAV to the final location at the $(t-1)$-th and $t$-th time steps, respectively; and $\zeta$ is a tunable parameter during the learning process. The second element on the right-hand side of (11) encourages the UAV to fly towards the final location.

- **Policy:** A projection from the state space, $\mathcal{S}$, to the action space $\mathcal{A}$ is referred to as a policy, $\mu : \mathcal{S} \to \mathcal{A}$, a distribution $\mu(a|s) = \Pr(a_t = a|s_t = s)$ over state $s \in \mathcal{S}$.
- **Experience:** The experience, defined as $e_t = (s_t, a_t, r_t, s_{t+1})$, is stored in an experience replay memory $\boldsymbol{R}$.

The UAV experiences state $s_t$, performs action $a_t$, receives reward $r_t$, and turns to state $s_{t+1}$. A policy $a_t = \mu(s_t)$ maps state $s_t$ to a possible action. The UAV chooses the policy that maximizes the cumulative reward $R_t = \sum_{n=t}^N \gamma^{n-t} r_t$. Here, $\gamma \in (0, 1)$ gives the discount factor. Given $s_t$, $a_t$, and $\mu$, the Q-function evaluates $R_t$ by

$$Q_\mu(s_t, a_t) = \mathbb{E}_\mu[R_t|s_t, a_t]. \tag{12}$$

The action-value function, $Q_\mu(s_t, a_t)$, follows the Bellman Expectation Equation:

$$Q_\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim \mathcal{E}} \left[r_t + \gamma \mathbb{E}_{a_{t+1} \sim \mu} \left[Q_\mu(s_{t+1}, a_{t+1})\right]\right]. \tag{13}$$

Here, $\mathcal{E}$ stands for the environment the UAV experiences.

It is generally challenging to directly use an RL algorithm to solve the continuous-space, finite-horizon MDP and determine the Q-value, $Q(s_t, a_t)$, due to the continuous state and action spaces. This paper puts forth a new DDPG-based algorithm to control the UAV's trajectory and configure the RIS, as delineated in the following subsection.

### B. Actor-Critic Framework-Based DDPG

The DDPG-based network uses four DNN approximators, including training-actor and training-critic networks, and target-actor and target-critic networks, as shown in Fig. 2. The training-actor network with parameters $\theta_a$, denoted as $\mu(s_t; \theta_a)$, gives an approximate policy of the UAV and produces the actions. The training-critic network with parameters $\theta_c$, denoted as $Q_\mu(s_t, a_t; \theta_c)$, estimates the action-value function concerning the actions created in the training-actor network [35]. The target-actor network with parameter $\theta_a'$, represented by $\mu'(s_t; \theta_a')$, and the target-critic networks with parameter $\theta_c'$, represented by $Q_{\mu'}'(s_t, a_t; \theta_c')$, generate the target Q-value for training the training-actor and training-critic networks.

Fig. 2. The proposed DDPG-based framework with a training network and a target network, each comprising an actor network and a critic network. The experience replay buffer gives batches of samples of state transitions for training and updating the networks.



Fig. 3. The proposed TD3-based framework with an actor network comprising an actor and a target-actor, and a critic network comprising two critics and two target-critics. The experience replay buffer gives batches of samples of state transitions for training and updating the networks.

The DDPG network uses the deterministic policy gradient (DPG) theorem [35] to refresh $\theta_a$, $\theta_c$, $\theta_a'$, and $\theta_c'$. It produces actions in an actor-critic setting. Additionally, the adoption of a target network (i.e., the target-actor and target-critic networks) helps prevent unstable learning, as opposed to using only a training network (with a training-actor and a training-critic network) [38].

The UAV inputs state $s_t$ into the training-actor network. Using the DPG theorem [35], the network generates the strategy by projecting the state to an action in a deterministic fashion. This network approximates the agent's policy function and selects action $a_t$. A noise is added to $a_t$ to balance between new and known actions, resulting in an output action $a_t = \mu(s_t; \theta_a) + \mathcal{N}_t$. Herein, $\mathcal{N}_t$ is a random noise process with a normal distribution. The agent is rewarded with $r_t$ and transitions to state $s_{t+1}$. Then, it stores the experience $(s_t, a_t, r_t, st+1)$ in $\boldsymbol{R}$.

The training-critic network evaluates the action-value function $Q_\mu(s_t, \mu(s_t; \theta_a); \theta_c)$ of the selected action $a_t$. By using a random sample from the replay memory $\boldsymbol{R}$, the network ap-

proximates the action-value function as $Q_\mu(s_i, \mu(s_i; \theta_a); \theta_c)$. We take $J(\theta_a)$ to be the probability distribution of the parameter $\theta_a$. The training-actor network is adjusted in the direction that improves the strategy the most rapidly, i.e., in the direction of the gradient of $J(\theta_a)$ with respect to (w.r.t.) $\theta_a$ [35]:

$$\nabla_{\theta_a} J(\theta_a) = \mathbb{E}_{s \sim \rho^\mu} \left[ \nabla_{\theta_a} Q_\mu(s_t, \mu(s_t; \theta_a); \theta_c) \right] \quad (14a)$$

$$= \mathbb{E}_{s \sim \rho^\mu} \left[ \nabla_{\theta_a} \mu(s_t; \theta_a) \nabla_a Q_\mu(s_t, a; \theta_c)|_{a = \mu(s_t; \theta_a)} \right], \quad (14b)$$

where (14b) uses the chain rule; $\rho^\mu$ provides a discounted state distribution of $\mu(s_t; \theta_a)$ [36]; $\nabla_{\theta_a} \mu(s)$ gives the gradient of the training-actor network $\mu(s)$ w.r.t. $\theta_a$; $\nabla_a Q_\mu(s_t, a; \theta_a)$ provides the gradient of $Q_\mu(s_t, a; \theta_a)$ w.r.t. $a$.

By randomly drawing $N_{batch}$ sampled historical transitions from $\boldsymbol{R}$, the gradient $\nabla_{\theta_a} J(\theta_a)$ is approximated by

$$\nabla_{\theta_a} J(\theta_a) \approx \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \left[ \nabla_{\theta_a} \mu(s_i) \nabla_a Q_\mu(s_i, a; \theta_c)|_{a = \mu(s_i)} \right]. \quad (15)$$

The training-actor network parameter, i.e., $\theta_a$, is refreshed based on the gradient ascent [39]

$$\theta_a \leftarrow \theta_a + \eta_a \nabla_{\theta_a} J(\theta_a)$$
$$\approx \theta_a + \frac{\eta_a}{N_{batch}} \sum_{i=1}^{N_{batch}} \left[ \nabla_{\theta_a} \mu(s_i) \nabla_a Q_\mu(s_i, a; \theta_c)|_{a=\mu(s_i)} \right]. \tag{16}$$

Here, $\eta_a$ specifies the learning rate of the training-actor network.

The training-critic network is refreshed through minimizing the following loss function:

$$L(\theta_c) = \mathbb{E}_{s_t \sim \rho^\mu, a_t \sim \mu(s_t; \theta_a)} \left[ (Q_\mu(s_t, a_t; \theta_c) - y_t)^2 \right]. \tag{17}$$

Here, $y_t = r_t + \gamma Q'_{\mu'}(s_{t+1}, \mu'(s_{t+1}; \theta'_a); \theta'_c)$ is the target Q-value produced by the target network under the transition $(s_t, a_t, r_t, s_{t+1})$. Here, the parameters of the target-actor and target-critic networks, $\theta'_a$ and $\theta'_c$, are the respective decayed copies of $\theta_a$ and $\theta_c$.

With $N_{batch}$ randomly sampled transitions, the loss function, $L(\theta_c)$, is approximately evaluated by

$$L(\theta_c) \approx \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \left[ (Q_\mu(s_i, \mu(s_i; \theta_a); \theta_c) - y_i)^2 \right], \tag{18}$$

where $y_i = r_i + \gamma Q'_{\mu'}(s_{i+1}, \mu'(s_{i+1}; \theta'_a); \theta'_c)$ gives the approximate target Q-value that the target network generates upon $N_{batch}$ transitions sampled at random. By differentiating $L(\theta_c)$ w.r.t. $\theta_c$, the gradient is attained:

$$\nabla_{\theta_c} L(\theta_c) \approx \frac{2}{N_{batch}} \sum_{i=1}^{N_{batch}} [(Q_\mu(s_i, \mu(s_i; \theta_a); \theta_c) - y_i) \tag{19}$$
$$\times \nabla_{\theta_c} Q_\mu(s_i, \mu(s_i; \theta_a); \theta_c)].$$

The training-critic network parameter, $\theta_c$, is refreshed by utilizing the stochastic gradient descent method [39].

The target-actor and target-critic networks are refreshed based on the training-actor and training-critic networks:

$$\theta'_a \leftarrow \tau_a \theta_a + (1 - \tau_a) \theta'_a,$$
$$\theta'_c \leftarrow \tau_c \theta_c + (1 - \tau_c) \theta'_c, \tag{20}$$

where $\tau_a$ and $\tau_c$ are the decaying rates for the training-actor and training-critic networks, respectively.

### C. Twin Delayed DDPG (TD3)

TD3 is one of the latest extensions of DDPG and consists of a training network and a target network, where the training network is made of a training-actor and two training-critic networks, and the target network comprises a target-actor and two target-critic networks, as shown in Fig. 3. TD3 addresses the Q-value overestimation problem of the DDPG algorithm by incorporating *three improvements* over the classical DDPG model, namely, *clipped double-Q learning*, *target policy smoothing*, and *delayed policy update* [40], [41].

- *Clipped double-Q learning:* TD3 contains two training-critic and target-critic networks to produce two Q-values. The lesser of the two is used to evaluate the target Q-value in the Bellman error loss function. Specifically,

$Q_\mu(s_t, a_t; \theta_c)$ in the DDPG is replaced by $Q_\mu(s_t, a_t) = \min \{Q_1(s_t, a_t; \theta_1), Q_2(s_t, a_t; \theta_2)\}$ in the TD3.

- *Target policy smoothing:* TD3 perturbs actions produced by the target-actor network (i.e., "target action") with noises and smooths the corresponding Q-function values to enhance the resistance of the policy against erroneous Q-functions. The smoothed target action is written as

$$a'_t = \text{clip}\left( \mu'(s_{t+1}; \theta'_a) + \text{clip}\left( \epsilon', -\sigma_m^2, \sigma_m^2 \right), a_{\min}, a_{\max} \right). \tag{21}$$

Here, the noise $\epsilon'$ is taken at random from a Gaussian distribution with zero mean and variance $\sigma_a^2$, i.e., $\epsilon' \sim \mathcal{N}(0, \sigma_a^2)$; and $\sigma_m^2$ is the maximum exploration noise supported by the environment. In contrast, the DDPG model does not add noises towards target actions.

- *"Delayed" policy update:* The training-actor and target-actor networks (i.e., policies) are refreshed less frequently than the training-critic and target-critic networks. For example, it was recommended in [40] that the training-actor and target-actor networks are refreshed after the training-critic and target-critic networks are refreshed twice in TD3. In contrast, the classical DDPG model refreshes its train-actor and target-actor networks and train-critic and target-critic networks at the same pace.

### V. PERFORMANCE EVALUATION

We carry out extensive experiments in Python to evaluate the proposed approach. The location of the jammer is $q_J = [-25, -25, 0]^T$ m. The UAV's initial and final locations are $q_0 = [-200, -100, 5]^T$ m and $q_F = [100, 60, 50]^T$ m. The RIS has $N = 5 \times 4 = 20$ (or $N = 5 \times 8 = 40$) reflecting elements, and the reference point is $q_R = [50, 50, 30]^T$ m. The scheduling horizon is $T = 30$ s with each time slot being $\delta = 0.1$ s. The other parameters concerning the system model are collated in Table I.

### A. Experiment Settings

The proposed DDPG network is composed of actor networks implemented using fully connected neural networks (FCNNs) with three hidden layers and learning rates of $10^{-4}$. The first, second, and third layers of the actor networks have 64, 128, and 64 neurons, respectively. The output layer implements the $\tanh(\cdot)$ activation function to bound the output actions within $[-\pi, \pi)$ for the RIS configuration and $[-2, 2]$ m/s² for the UAV control. Additionally, the paper utilizes critic networks that employ FCNNs with two hidden

TABLE II
THE HYPERPARAMETERS OF THE PROPOSED DDPG AND TD3
ALGORITHMS

| Parameter | Value |
| --- | --- |
| Reduction coefficient for upcoming reward, $\gamma$ | 0.99 |
| Training coefficient for actor and critic networks, $\eta_a$, $\eta_c$ | $1 \times 10^{-4}$ |
| Declining coefficient for actor and critic networks, $\tau_a$, $\tau_c$, $\rho_\tau$ | $5 \times 10^{-3}$ |
| Capacity for experience repetition | $1 \times 10^5$ |
| Quantity of episodes, $T_{ep}$ | 3000 |
| Total steps per episodes, $T_s$ | 300 |
| Quantity of experiences in a mini-batch, $N_{batch}$ | 128 |
| Variance of the exploration noise, $\sigma_e^2$ | 0.2 |
| Delayed policy update interval (TD3) | 2 |
| Variance of the policy noise (TD3), $\sigma_a^2$ | 0.2 |
| Largest value of the Gaussian noise (TD3), $\sigma_m^2$ | 0.5 |

layers and learning rates of $10^{-3}$. Both hidden layers utilize the Rectified Linear Unit (ReLU) activation functions with 64 neurons in the first layer and 128 neurons in the second layer. The DDPG actor policy is trained using additive noise $\mathcal{N}$, which is sampled from a complex Gaussian noise distribution with zero mean and variance 0.2.

The proposed TD3 network is built upon the DDPG network. It includes two duplicates of the training-critic and target-critic networks; see Fig. 3. Similar to the DDPG network, the actor in the TD3 network is trained using exploration noise that is drawn from a complex Gaussian distribution with zero mean and variance 0.2. Additionally, the target-actor in the TD3 network is smoothed using policy noise that is drawn from a complex Gaussian distribution with zero mean and variance 0.2. The maximum exploration noise is set to 0.5, and the actor networks are refreshed every two steps. The TD3 improves the DDPG by providing faster and smoother convergence, which is especially beneficial for larger RISs.

The hyperparameters of the proposed DDPG and TD3 networks are summarized in Table II. The DDPG and TD3 networks are trained on a server equipped with a NVIDIA Tesla P100 SXM2 16GB GPU.

**Baseline 1:** This baseline applies TD3 to UAV flight path planning in the absence of the RIS, referred to as "without RIS". The TD3 algorithm learns the UAV's trajectory solely based on the received data rate at the UAV without CSI involving the RIS or jammer.

**Baseline 2:** This baseline decouples the UAV's trajectory plan from the RIS configuration by first using a TD3-based algorithm to optimize the UAV's trajectory given the RIS configuration, and then maximizing the signal-to-noise ratio (SNR) at each time slot using the Dinkelbach method under the assumption of perfect and instantaneous CSI for all involved channels. The Dinkelbach method is used to reformulate the SNR maximization problem as a fractional program defined as $F(\gamma_U^t) = \min_{\Theta_t} f(\Theta_t) - \gamma_U^t g(\Theta_t)$, s.t. $-\pi \leq \theta_n^t < \pi, \forall n, t$, where $\gamma_U^t = \frac{f(\Theta_t)}{g(\Theta_t)}$ and $\Theta_t$ gives the RIS configuration. Given $\gamma_U^t$, the fractional program can be reorganized as a quadratic program with a unit-modulus constraint and solved using manifold optimization. The value of $\gamma_U^t$ is refreshed based on the resultant $\Theta_t$. This process is repeated until convergence,



Fig. 4. The per-episode and average rewards of the proposed DDPG and TD3 algorithms. Fig. 4(a) plots the proposed TD3 scheme when $N = 20$; Fig. 4(b) plots the proposed TD3 scheme when $N = 40$; Fig. 4(c) plots Baseline 2 when $N = 20$; Fig. 4(d) plots the proposed DDPG scheme when $N = 20$; Fig. 4(e) plots the proposed DDPG scheme when $N = 40$; and Fig. 4(f) plots Baseline 1 "without RIS".

and the convergent value of $\gamma_U^t$ is output [42].

### B. Results of Policy Learning

Fig. 4 plots the per-episode and average rewards of the proposed and baseline algorithms for $N = 20$ and $N = 40$. The average reward for the $i$-th training episode, denoted by $\bar{r}_i$, is evaluated as $\bar{r}_i = \frac{1}{i} \sum_{j=1}^{i} r_j$, where $i = 1, \cdots, T_{ep}$, and $r_j$ is the step reward for the $j$-th training episode; see (11).

Fig. 4 shows that the average reward gradually increases, as the UAV control policy adapts to a randomly generated target trajectory in each episode. The proposed TD3 algorithm outperforms the baselines, and achieves its maximum reward at the 654th episode for $N = 20$, and at the 477th episode for $N = 40$. The proposed DDPG algorithm reaches its maximum reward at the 1,607th episode when $N = 20$, and at the 2,995th when $N = 40$. Baseline 1 without RIS reaches its maximum reward at the 280th episode. Baseline 2 reaches its maximum reward at the 1,752nd episode when $N = 20$. The fast convergence of Baseline 2 is due to its substantially smaller action space of only UAV accelerations resulting from an unrealistic assumption of perfect and instantaneous CSI of

Fig. 5. 3D UAV trajectory, where the green and red dots are the UAV initial location and its expected destination, the orange and yellow triangles denote the locations of the BS and Jammer, and the blue square is the RIS reference point.



Fig. 6. Projection of the UAV trajectory on the $x$-$y$ plane, with the green and red dots being the UAV initial and final locations, the orange and yellow triangles being the locations of the BS and Jammer, and the blue square being the RIS reference point.

all involved channels. In general, the TD3 converges faster and more smoothly than the DDPG. Yet, it undergoes less smooth changes in the per-episode reward when the action space is smaller, i.e., $N = 20$, This is because the TD3 has two critic networks for both the training and target networks (c.f. Fig. 3), which could incur higher complexity and more randomness when training, especially when the action space is smaller. In contrast, DDPG is suitable for training tasks that are less complicated and have relatively smaller action spaces.

### C. Test Results of Learned Policy

Using the learning results obtained in Section V-B, we test the proposed DDPG-based and TD3-based algorithms for $N = 20$ and $40$ at the RIS, as well as the baseline schemes for $N = 20$ at the RIS for comparison. 500 testing episodes are conducted, each consisting of 300 steps. During testing, no exploration noise is added. The proposed algorithms and baselines are evaluated in terms of the 3D UAV trajectory and received data rate. Fig. 5 shows the 3D trajectory of the UAV, while Fig. 6 illustrates the trajectory in the $x$-$y$ plane and along the $z$-axis. These figures demonstrate that the UAV is able to adapt its control policy to the anti-jamming communication and successfully reach the destination.

Fig. 7 shows the received data rate of the UAV as the mission duration increases. The results are based on an average of 500 independent testing episodes, with error bars representing the associated uncertainty. It is observed that the received data rate increases with the mission duration under all considered algorithms, and the use of an RIS improves the received data rate. The proposed TD3 and DDPG algorithms give slightly lower data rates than Baseline 2, but operate without the CSI involving the RIS or the jammer, demonstrating their ability

to adapt to system changes. Additionally, the proposed TD3 algorithm achieves a substantially higher data rate than the proposed DDPG algorithm, particularly for larger numbers of RIS elements. This is attributed to the faster convergence and better convergent control policy of TD3 compared to DDPG, as previously shown in Fig. 4.

Fig. 8 illustrates the distance between the UAV and its expected destination at the end of the mission. The results show that as the mission duration increases, the distance decreases for all algorithms. Without the use of the RIS, the UAV is unable to reach its expected final location as it must remain close to the BS to maintain a sufficient data rate. However, the proposed algorithms, such as TD3, enable the UAV to get closer or reach its final location by adjusting the RIS to enhance the desired signals, weaken the jamming signals and extend the effective BS-UAV transmission range.

It is worth noting that as the mission duration increases, DDPG can increasingly approach TD3 in the average achievable data rate. This indicates that DDPG is suitable for a less constrained problem setting where there is sufficient time for the UAV to maneuver and explore its action space. In this case, DDPG can be a suitable solution, as it can benefit from its simpler network architecture than TD3. In contrast, TD3 demonstrates significant gains over DDPG when the time

Fig. 7. The received data rate of the UAV vs. the mission duration averaged over 500 independent testing runs.



Fig. 8. The finishing distance between the UAV and its destination at the end of the mission with the increase of the mission duration.



Fig. 9. Cumulative distribution function (CDF) of the distance from the UAV to the final location when the mission duration is $T = 30$ and $40$ s.



Fig. 10. The UAV's data rate vs. its distance to the expected destination, where $T$ ranges from 5 to 40 seconds.

constraint is more stringent. In other words, TD3 suits better under a shorter mission duration, since it has a more complex network structure and can generate more randomness to test the action space more extensively for better solutions.

Fig. 9 plots the cumulative distribution function (CDF) of the distance between the UAV and its expected destination at the end of the mission, for mission durations of $T = 30$ and $40$ s. The results show that as the number of RIS elements (i.e., $N$) or the mission duration (i.e., $T$) increases, the proposed TD3 algorithm can get closer to or reach the destination more frequently, and performs significantly better than the case without an RIS. While perfect CSI is important for UAV trajectory planning, as seen in Baseline 2 for $N = 20$, the TD3 algorithm can produce equally effective trajectories without CSI by utilizing a larger RIS with more elements (as seen in TD3 for $N = 40$). On the other hand, the DDPG algorithm appears to suffer from overfitting, as the distance between the UAV and its expected destination has little dispersion. This

indicates that DDPG is more prone to overestimating the $Q$-value function for a small number of possible actions, leading to a noisy gradient for policy refreshes and less effective UAV trajectories and lower data rates.

## VI. CONCLUSION

This paper developed a new DRL-driven framework for the trajectory planning and RIS-assisted jamming rejection for a UAV-borne IoT platform. The DDPG model and its enhancement, TD3, were designed to allow the UAV to learn its trajectory and the RIS configuration only based on its received data rate, eliminating the need of CSI for learn-

ing. Extensive simulations showed that the proposed DRL algorithms offer the UAV reliable resistance against jamming. The TD3 algorithm converges faster and more smoothly than the DDPG algorithm. It also demonstrates robustness against different locations of the jammer. This is particularly important due to the difficulty in locating the jammer in practice.

## REFERENCES

[1] B. Bera, A. K. Das, S. Garg, M. Jalil Piran, and M. S. Hossain, "Access control protocol for battlefield surveillance in drone-assisted IoT environment," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2708–2721, Feb. 2022.

[2] L. Liu, A. Wang, G. Sun, and J. Li, "Multiobjective optimization for improving throughput and energy efficiency in UAV-enabled IoT," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20763–20777, Oct. 2022.

[3] K. Li et al., "Energy-efficient cooperative relaying for unmanned aerial vehicles," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1377–1386, 2016.

[4] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

[5] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "On-board deep q-network for uav-assisted online power transfer and data collection," *IEEE Trans. Veh. Tech.*, vol. 68, no. 12, pp. 12215–12226, 2019.

[6] S. Hu, W. Ni, X. Wang, A. Jamalipour, and D. Ta, "Joint optimization of trajectory, propulsion, and thrust powers for covert UAV-on-UAV video tracking and surveillance," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1959–1972, Jan. 2021.

[7] X. Yuan et al., "Secrecy rate analysis against aerial eavesdropper," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7027–7042, Oct. 2019.

[8] S. Hu, Q. Wu, and X. Wang, "Energy management and trajectory optimization for UAV-enabled legitimate monitoring systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 142–155, Jan. 2021.

[9] Y. Dang, C. Benzaïd, B. Yang, T. Taleb, and Y. Shen, "Deep-ensemble-learning-based GPS spoofing detection for cellular-connected UAVs," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25068–25085, Dec. 2022.

[10] K. Li, R. C. Voicu, S. S. Kanhere, W. Ni, and E. Tovar, "Energy efficient legitimate wireless surveillance of uav communications," *IEEE Trans. Veh. Tech.*, vol. 68, no. 3, pp. 2283–2293, 2019.

[11] H. Lei et al., "Safeguarding UAV IoT communication systems against randomly located eavesdroppers," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1230–1244, Feb. 2020.

[12] Z. Na, Y. Liu, J. Shi, C. Liu, and Z. Gao, "UAV-supported clustered NOMA for 6G-enabled internet of things: Trajectory planning and resource allocation," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15041–15048, Oct. 2021.

[13] X. Yuan et al., "Secrecy performance of terrestrial radio links under collaborative aerial eavesdropping," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 604–619, 2020.

[14] B. Duo, Q. Wu, X. Yuan, and R. Zhang, "Anti-jamming 3D trajectory design for UAV-enabled wireless sensor networks under probabilistic LoS channel," *IEEE Trans. Veh. Tech.*, vol. 69, no. 12, pp. 16288–16293, Dec. 2020.

[15] J. Peng, Z. Zhang, Q. Wu, and B. Zhang, "Anti-jamming communications in UAV swarms: A reinforcement learning approach," *IEEE Access*, vol. 7, pp. 180532–180543, Dec. 2019.

[16] Z. Li et al., "UAV networks against multiple maneuvering smart jamming with knowledge-based reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12289–12310, Aug. 2021.

[17] C. Huang et al., "Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 118–125, Oct. 2020.

[18] Y. Cao, T. Lv, Z. Lin, and W. Ni, "Delay-constrained joint power control, user detection and passive beamforming in intelligent reflecting surface-assisted uplink mmWave system," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 2, pp. 482–495, Jun. 2021.

[19] C. Sun, W. Ni, Z. Bu, and X. Wang, "Energy minimization for intelligent reflecting surface-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, To appear, 2021.

[20] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 3313–3351, May 2021.

[21] Y. Liu et al., "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Commun. Surv. Tut.*, vol. 23, no. 3, pp. 1546–1577, 3rd Quart. 2021.

[22] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019.

[23] ——, "Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1838–1851, Mar. 2020.

[24] S. Xu, J. Liu, and Y. Cao, "Intelligent reflecting surface empowered physical-layer security: Signal cancellation or jamming?" *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1265–1275, Jan. 2022.

[25] Z. Peng et al., "Deep reinforcement learning for RIS-aided multiuser full-duplex secure communications with hardware impairments," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21121–21135, Nov. 2022.

[26] Y. Xu et al., "Computation capacity enhancement by joint UAV and RIS design in IoT," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20590–20603, Oct. 2022.

[27] S. Li, B. Duo, X. Yuan, Y. Liang, and M. Di Renzo, "Reconfigurable intelligent surface assisted UAV communication: Joint trajectory design and passive beamforming," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 716–720, May 2020.

[28] H. Wang, R. P. Liu, W. Ni, W. Chen, and I. B. Collings, "Vanet modeling and clustering design under practical traffic, channel and mobility conditions," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 870–881, 2015.

[29] A. Ranjha and G. Kaddoum, "URLLC facilitated by mobile UAV relay and RIS: A joint design of passive beamforming, blocklength, and UAV positioning," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4618–4627, Mar. 2021.

[30] X. Liu, Y. Liu, and Y. Chen, "Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2042–2055, Jul. 2021.

[31] X. Mu, Y. Liu, L. Guo, J. Lin, and H. V. Poor, "Intelligent reflecting surface enhanced multi-UAV NOMA networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3051–3066, Oct. 2021.

[32] C. Sun, W. Ni, and X. Wang, "Joint computation offloading and trajectory planning for UAV-assisted edge computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5343–5358, Aug. 2021.

[33] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, Jun. 2019.

[34] H. Lu, Y. Zeng, S. Jin, and R. Zhang, "Aerial intelligent reflecting surface: Joint placement and passive beamforming design with 3D beam flattening," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4128–4143, Jul. 2021.

[35] D. Silver et al., "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, 2014, pp. 387–395.

[36] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. ICLR (Poster)*, 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[37] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed machine learning for wireless communication networks: Techniques, architectures, and applications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1458–1493, 3rd Quart., 2021.

[38] Y. Hou, L. Liu, Q. Wei, X. Xu, and C. Chen, "A novel DDPG method with prioritized experience replay," in *Proc. IEEE Int. Conf. Systems, Man, Cybernet. (SMC)*, 2017, pp. 316–321.

[39] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation." in *Proc. NIPS*, vol. 99.  Citeseer, 1999, pp. 1057–1063.

[40] S. Dankwa and W. Zheng, "Twin-delayed DDPG: A deep reinforcement learning technique to model a continuous movement of an intelligent robot agent," in *Proc. 3rd Int. Conf. Vision, Image, Signal Process.*, 2019, pp. 1–5.

[41] X. Yuan, S. Hu, W. Ni, R.-P. Liu, and X. Wang, "Joint user, channel, modulation-coding selection, and RIS configuration for jamming resistance in multiuser OFDMA systems," *IEEE Trans. Commun.*, Early access, Jan. 2023.

[42] W. Dinkelbach, "On nonlinear fractional programming," *Management science*, vol. 13, no. 7, pp. 492–498, 1967.